

## Software: Initial Overview

- Key tasks and decision points
- Organization: software subgroups & manpower
- Sample flowchart for full chain
- A couple of strawman plans
- Goals for this workshop

# Software Subgroups

## ① DAQ

- fast **readout** and **event builder**
- Betsy: **simulated DAQ output** for testing other programs

## ② Slowcontrol

- **CODA readout** of hardware monitors + scalers
- **online monitoring** including running analysis chain
- **synchronization** with event stream  
(e.g. reconstructed events + slowcontrol/scalers → **DSTs**)

## ③ Decoder & Reconstruction

- talk to **new DAQ** (whatever output format we decide on)
- **decoder**: mapping, alignment & calibration
- **reconstruction rewrite** in C for maintenance & rate limits
- must take **input from MC** (pre- or post-decoder?) for testing

## Software Subgroups

### ④ Monte Carlo

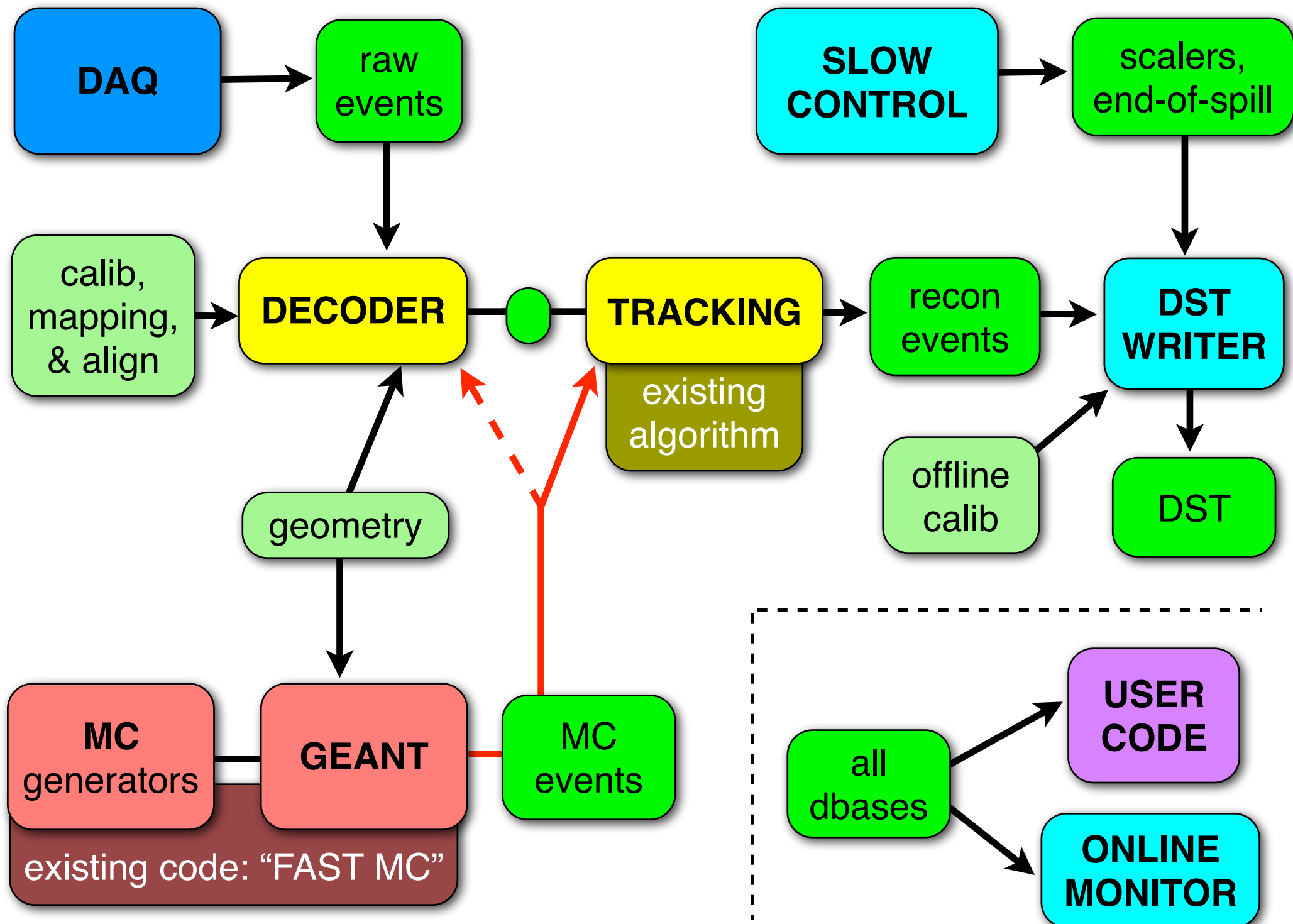
- **FastMC** and/or **GEANT4** development lines → **decide**
- urgently need both **background** & **signal** (physics) generators for design decisions
- MC output must **talk to decoder** and/or **tracking** code

### ⑤ Usercodes

- standard **ntuple/root-tree writers** and/or example codes
- get some **key macros** ready for online analysis

### ⑥ Documentation & code administration

- good **tutorials** and **example codes**
- database documentation
- **SVN repository** (now exists at UIUC) and **code czar**



## Record Structure

*Decision: how to organize **spill – event – track** level info*

```
smData = {  
  smTrack, smCluster, smLumi, smRICH,  
  glData = {  
    glTrack, glDAQ, glQuality, glHVtrip, glQualInfo, glBurstStat,  
    glDetector, glBeam, glTarget, glHE3, glABS, glUnpol, glTrkEffi,  
    glACE, glACEcnts, glOnline, gluDSTstat, glSpinGate, glTrigger }  
}
```

- **burst-level DSTs**: 1 record per 10-second burst = readout interval for scalers, lumi, slowcontrol
- **burst-level tables**: 1 row per record
- **track-level tables**: 1 row per track
- **other-level tables**: 1 row per calo cluster, lumi cluster, trigger# (prescale factors), ...
- **event structure** imposed by **DST writer** and **usercode library**: event# used to collect tracks & clusters into events via RDBM links

# Manpower for software development

## Online

- Academica Sinica & Ling-Tung Univ: DAQ + slowcontrol

## Offline

- Illinois
  - 25% Naomi
  - 100% grad student TBN (hopefully start this semester)
- Maryland
  - 10% Betsy
  - 50% Prabin (PhD, other 50% on target control)
  - 35% postdoc TBN (hopefully start this May)
- ACU
  - 10% Shon Watson → Monte Carlo
  - 40% undergrad(s) during summer / Xmas
- LANL
  - 10% Ming + Mike + Pat → supervising ACU student(s) during summer has worked very well

# Strawman Plan A: minimal

## *Essential software tasks*

- **new DAQ**, but write to same flat-file format as in 866
- **slowcontrol**: need readout, synching, and online monitoring
- **MC**: pick official MC(s) and MC czar(s) for bg & signal events
- **reconstruction: C rewrite** of 866 code
  - keep algorithm: thoroughly tested and optimized
  - need dynamic memory allocation to get around combinatoric limits on track finding in high-occupancy events
  - further development of legacy code unmanageable
- identify **minimal set of changes** needed to 866
  - database formats (flat files)
  - decoder
  - slowcontrol/scaler readout & synching w event stream
- **example codes** and **documentation**

## Strawman Plan B: excellent

### *Desired software changes for flexibility & ease of use*

- **MySQL databases** at all I/O points
  - dbase carries metadata → **flexible!** codes will be robust against ongoing database changes!
  - numerous **access tools** exist & are thoroughly tested
  - numerous **APIs** exist → can access MySQL databases from C, root, perl, UNIX shell, ...
  - all data & MC files in **common format** → use similar tools
- **event display** e.g. GEANT-based display of output from analysis chain, for use during data taking & offline analysis
- **analysis macros** for key analyses: ready-to-go and tested on MC before data taking
- **modular Monte Carlo** with support for multiple generators: feed physics/bg generators to same detector simulation



## Tasks we must complete at this workshop

- **Learn** 866 software!
- Upload running 866 codes to UIUC **SVN repository**
- Decide on **top-priority MC path(s)** to follow & associated go-to people for design decisions
- Hash out semi-final **software flowchart**
- Prepare **detailed task list** including
  - priorities (required vs desired changes)
  - people
  - draft schedule
- Decide on **database format & general structure** at all access points (incl MC and slowcontrol / scaler / end-of-spill)
- ... and hopefully hash out detailed **database design**